<http://www.diva-portal.org>

Postprint

This is the accepted version of a paper published in *Multimedia tools and applications*. This paper has been peer-reviewed but does not include the final publisher proof-corrections or journal pagination.

Citation for the original published paper (version of record):

Rossholm, A., Lövström, B. (2016)

A robust method for estimating synchronization and delay of audio and video for communication services.

Multimedia tools and applications, 75(1): 527-545

<http://dx.doi.org/DOI 10.1007/s11042-014-2306-6>

Access to the published version may require subscription.

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:bth-6407>

A robust method for estimating synchronization and delay of audio and video for communication services

Andreas Rossholm · Benny Lövström

Received: 22 December 2013 / Revised: 14 August 2014 / Accepted: 1 October 2014
© Springer Science+Business Media New York 2014

Abstract One of the main contributions to the quality of experience in streaming services or in two-way communication of audio and video applications is synchronization. This has been shown in several studies and experiments but methods to measure synchronization are less frequent, especially for situations without internal access to the application and independent of platform and device. In this paper we present a method for measuring synchronization skewness as well as delay for audio and video. The solution incorporates audio and video reference streams, where audio and video frames are marked with frame numbers which are decoded on the receiver side to enable calculation of synchronization and delay. The method has been verified in a two-way communication application in a transparent network with and without inserting known delays, as well as in a network with 5 and 10 % packet loss levels. The method can be used for both streaming and two-way communication services, both with and without access to the internal structures, and enables measurements of applications running on e.g. smartphones, tablets, and laptops under various conditions.

Keywords Lip sync · Synchronization · Delay · QoE · Video streaming · Video conferencing

1 Introduction

The widespread use of video communication and video streaming in a growing number of application areas gives emphasis to the topics of Quality of Service (QoS) and Quality of Experience (QoE). Several factors influence the QoE, including source quality, encoding degradation, network behaviour, and decoder and rendering performance. The methods to measure the quality, especially in a quantitative way, is a large research topic having many

A. Rossholm · B. Lövström (✉)
Blekinge Institute of Technology, 371 79 Karlskrona, Sweden
e-mail: benny.lovstrom@bth.se

A. Rossholm
e-mail: a.rossholm@gmail.com

aspects. One of the most important factors for the audiovisual quality is synchronization of audio and video. This is of high importance when it comes to streaming services as well as real time applications for one-way or two-way communication of audio and video. Several studies have been published focused on both the effects of audio and video skewness and on reference models to handle synchronization in different ways, at IP level as well as application level [1, 7, 23, 25]. It is shown that viewers perceive audio and video to be synchronized with an audio-to-video skew up to about 80 ms but also that there is a higher tolerance for video ahead of audio than vice versa. Further, the type of content as well as the quality of the video, e.g. video resolution, quantization level, and frame rate are also impacting the perceived skewness [6, 22]. In the case of real time two-way communication another important contribution to the quality of experience is delay. The delay can impact synchronization especially in the case of separate audio and video streams, but also the delay itself has impact on the QoE of users sharing information instantly and continuously [24]. It has been shown that a delay less than 100-150 ms is preferred and above 400 unacceptable, which also is stated in several specifications [5]. However, recent studies on speech have shown that interactivity has a big impact on the perceived quality and that people can adapt to the current situation [18] compared to quality scale used in [12].

To evaluate synchronization and delay for video communication applications like Microsoft's Skype, Google's Hangouts, Apple's FaceTime it is required that several issues are taken into account to get a complete evaluation and support different kind of scenarios, e.g. different network conditions, different platforms and devices running under different constraints with different operating systems. These requirements results in the need for a method that is robust to e.g. different packet losses and jitter as well as different compressions levels and CPU constraints. In this paper a novel, robust, out-of-service measurement method is presented, and since it is a standalone application it is supporting both different platforms and different devices. The method can be used for measuring both synchronization between audio and video and delay. The method uses a pre-generated test signal fed into the sender's audio and video input, and on the receiver side the audio and video output are captured and processed. However, it would also be possible to apply the pre-stored frame codes to the incoming audio and video signals on the sender side to construct an in-service measurement which would enable synchronization measurement at the receiver side.

The paper is organized as follows. A technical background is given in Section 2, and in Section 3 published work related to this paper is discussed. In Section 4 the proposed method is presented including a more detailed description of the audio and video stamps and how the detection of the reference signal is performed. In Section 5 a description of a proof of concept is given and in Section 6 the results are presented. Finally, in Section 7 summary and conclusions are given.

2 Technical background

When a video sequence with related audio content is streamed over a network there are two main systems that can cause audio and video to be out of sync, or skewed. One is the transport over the network, and the other is the sending and receiving equipment which usually processes audio and video separately. The transport employed in most streams today is packet-based, using Internet Protocol (IP) with for e.g. User Datagram Protocol (UDP) and Real-time Transport Protocol (RTP) or Transmission Control Protocol (TCP), where

audio and video can be handled in different paths as well as multiplexed. In transmission of audio and video data over a network a number of trade-off decisions are made, such as between getting acceptable delays or have a low packet loss rate and jitter, or considerations regarding bitrates, frame-rates and resolution. This all affects the quality the user finally experiences, and among the parameters affecting the experience are delay and synchronization. In addition, even if there is a big impact from the transport on delay and synchronization, also acquisition, compression, transmission, and reconstruction must be included when evaluating the impact since all of these stages will to some extent have impact on the accumulated end-to-end delay and audio and video skewness. This means that even if the data stream would be unencrypted it would not be possible to only use significant information from used packets, e.g. RTP [8] or TCP, for reliable measurements.

There are several recommendations for the accuracy of synchronization and delay, varying for different user scenarios and also between the recommending bodies. In the television context ITU-T recommended synchronization thresholds in J.100 [10], which are 20 ms for audio lead and 40ms for audio lag. This recommendation provides a fixed figure for all content types and is intended to ensure that synchronization errors remain imperceptible. For real-time two-way low bitrate video communication, ITU recommends the asynchrony to be less than 100 ms. Further ITU sets the preferred one-way end-to-end delay to be 100 ms and the upper limit to be 400 ms [13]. For the same application ETSI [4] recommends less than 40 ms skew. For one-way broadcasting ITU recommends in [9] the skew to be less than 185 ms when video arrives first, and less than 90 ms when audio arrives first. This recommendation does not specify any preferred delay for the broadcasting scenario. For videophone ITU recommends a skew up to about 80 ms and end-to-end delay to be 150 ms with the upper limit 400 ms [11]. A summary of the presented recommendations is given in Table 1.

Table 1 Table showing recommendations for skew and one-way delay in different user scenarios and by different recommendation bodies

Rec. body	User scenario	Rec. nr.	Skew [ms]	Delay [ms]	Ref
ITU-T	Television	J.100	<20 (audio lead) <40 (audio lag)	–	[10],1990
ITU-T	Two-way video communication	H-series Supp. 1	<100	<100 (pref.) <400 (limit)	[13],1999
ETSI	Point-to-point videotelephony	ETR 297	<20 (audio lead) <40 (audio lag)	<800 (round trip)	[4],1996
ITU-R	Broadcasting	BT.1359-1	<90 (audio lead) <185 (audio lag)	–	[9],1998
ITU-T	Videophone	G.1010	<80	<100 (pref.) <400 (limit)	[11],2001

3 Related work

The impact of synchronization and delay on both QoE and QoS is well documented. Nevertheless, means or methods to measure this under real conditions, with a generic setup without limitations regarding e.g. platforms, operating systems (OS), devices and applications, is a topic that has not been published to the same extent. The topic can be seen from a broader network perspective or from the perspective of the communication application. In [21] peer-to-peer media streaming is discussed and the impact of delay and synchronization on QoS is addressed. Also, in the growing field of supporting different multimedia applications over heterogeneous networks the topic of delay and synchronization is addressed, where also the impact from the devices are taken into account [26, 27]. However, even if the frameworks addressing these topics and enabling several parts from a network and QoS perspective, there are still several impediments remaining when an evaluation or analysis should be performed. This comes mainly from the proprietary nature of the majority of applications resulting in lack of access to, and understanding of, protocol and system parameters which make insight and analysis problematic [21]. Further, as a result of today's tremendous dissemination of applications and platforms these different capabilities have a significant impact on the end result.

Methods for measuring synchronization and delay from the application perspective can be divided into two categories, out-of-service and in-service measurements, where out-of-service is also denoted as black-box tests. The difference is whether the method requires access to the application to enable the test method, which is the case for in-service methods, or if the method can be applied from the outside without any interaction or impacts of the application. Different out-of-service methods have been presented. A user centric measurement is presented in [14] enabling video delay measurements using QR codes, which was also further developed in [15] where audio measurements were added and thereby enabling synchronization measurements. A capture-to-display latency and frame rate estimation was presented in [2] where barcodes were used in a JAVA application running on the sender device. These methods enable measurements but in a limited range of scenarios. Also, the methods may affect the application resulting in an altered behaviour. Using QR codes or barcodes can, besides giving a complex decoding process, be limited in supporting e.g. high compression, small displays, and decreased frame resolution. All these limitations will reduce the scope of possible test scenarios.

In the area of in-service methods the use of watermarking or feature extraction has been suggested in e.g. [16, 19]. These in-service methods are usually limited by lack of robustness and they require sending of meta-data including relative temporal alignment information. A scenario with packet loss or fluctuating network bandwidth, resulting in temporal discontinuities or spatial fidelity change in the decoded sequences, can result in complications when it comes to extracting the features and to be able to preserve the correlation in a window of several seconds. Also, since in-service solutions require processing of the data inside the application this makes them unusable for evaluation of a not accessible application, e.g. a proprietary solution, since it requires the possibility to manipulate the application.

4 Proposed method

The main idea of the proposed method is to measure the audio and video delay and synchronization by marking the audio and video stream, before encoding, on the sender side with a signature or stamp in form of a frame number. This signature or stamp is designed

not to be corrupted in the encoding/decoding process or the transport. After decoding of the received audio and video stream these stamps are detected and decoded and used together with time information to calculate audio/video delay and synchronization. The frame number is generated by a binary code of M bits resulting in 2^M original numbers, which is coded with Gray codes [20] to reduce the impact on the video processing as well as being robust to different resolutions and compression levels. Further, with a frame rate of F_r fps this gives a maximum delay or skewness that can be detected of $2^M/F_r$ seconds.

The audio frames are marked with a sum of windowed sinusoids of different frequencies from a set of M sinusoidal base functions $b_1(n), \dots, b_M(n)$ with center frequencies $f_1, \dots, f_M$. The length of a base function is N samples. The set is chosen for each frame to form a code number for the frame, in line with the Gray coded frame numbering used for the video sequence. Additionally, to detect the beginning of a new audio frame an additional base function b_s or audio frame synchronization signal is added in the beginning of each frame. The audio signal is at the receiver side filtered in a bank of matched filters, in a similar way as in [3], having impulse responses $b_m(N - n), m = 1, \dots, M$, after which it can be decided which combination of base functions is present in a specific audio frame. In a corresponding manner the video frames are tagged by adding a pattern of black and white squares spatially distributed to each frame, where the pattern codes the frame number. In the proposed method a frame rate, F_r , of 15 fps is chosen to update frame number. This comes from a trade off between, on one side, as high refresh rate as possible of new measurements and, on the other side, sufficient support for different frame rates and camera capturing times for the video and reasonable length of the base functions for the audio.

This setup results in a robust measurement method both from a packet loss perspective and with respect to fluctuating bandwidth resulting in different compression levels and varying temporal or spatial resolution. It gives a robust solution with a possibility of new measurement every 1/15 second, and limited effect from fidelity change on the detection of the audio and video signatures and from limitations in the capturing device. The proposed method thereby enables the possibility to measure delay and synchronization in different kind of network and configurations. Also, any application, proprietary or not, can be tested.

4.1 The audio frame stamps

The base functions inserted in each audio frame will be sent through speech coders and decoders, and may be subject also to additional distortions. For the method to be robust the selection of base functions is essential. Due to the usual speech coder frequency range we have decided to keep the base functions in the frequency range of 700 to 2100 Hz. To get low correlation between the base functions we distribute them equidistant in this range, and choose long base functions. For a frame rate of F_r the number of samples in a frame will be F_s/F_r where F_s is the audio sampling rate. To have room for audio frame sync data in each frame and to have margins for consecutive frames, the length of the base functions were chosen as $N = F_s/(2 * F_r)$, i.e. half the frame length.

For the implementation in this paper the following values of the parameters F_s , F_r , N , and M were set. The audio channel sample rate is $F_s = 18000 \text{ Hz}$, and the frame rate is $F_r = 15 \text{ fps}$. This gives the base function length $N = 600$ samples. To be able to uniquely number up to 256 frames the number of base functions is set to $M = 8$. The separation of the base functions in frequency domain can be seen in Fig. 1 where the sinusoids have been windowed with rectangular window and Hanning window [17], respectively. In order to get

Fig. 1 Frequency functions of the base functions b_m , windowed by rectangular window (*top*) and Hanning window (*bottom*)

a robust detection system the interference between the base functions needs to be low. The figure shows that as expected the spectrum for each base function is widened by the use of Hanning window, but the better suppression of the side lobes makes Hanning window a better choice in order to reduce interference. The timing within the audio frames is seen in Fig. 2.

Fig. 2 Illustration of the position of the sum of base functions in the audio frames

Fig. 3 Code word entered into a video frame

4.2 The video frame stamps

The black and white squares representing the frame number is positioned in a video frame as shown in Fig. 3. To minimize the impact on the video encoder with discontinuities between two consecutive frames the frame number is coded with Gray codes resulting in that only one square is changed between each frame. The video sequence is updated with the same frame rate, $F_r = 15$ fps, and with the same frame number as the audio sequence. This gives a robust detection system since the squares are easy to detect and the impact from quantization and resolution change is small.

Fig. 4 Block diagram of the system measuring the audio/video delay and skew

Fig. 5 Block diagram of the detection of the audio frame number

4.3 The synchronization and delay detector

At the receiver side the audio and video is captured. For calculating synchronization the received audio signal $\hat{A}$ and the video signal $\hat{V}$ are used, and to enable delay measurements also the audio and video reference signals A and V are needed. A block diagram of the system at the receiver is shown in Fig. 4. There are two pre-processing blocks, Split, which is used to separate the audio and video signal, and Re-Sampling, which is applied to both audio and video signals to decimate the data in order to reduce the processing needed in subsequent steps. The block for detecting audio frame number in Fig. 4 is shown in detail in Fig. 5, and the block for detecting video frame number is shown in detail in Fig. 6.

Here it can be seen that the audio sequence is filtered with matched filters for both the synchronization signal and the base functions. The results from the synchronization filter are used in the frame splitting for further base function detection. These data are then

Fig. 6 Block diagram of the detection of the video frame number

digitized and converted from a Gray code to the actual frame number together with the corresponding time index captured from the synchronization match. For the video detection in Fig. 6 the video signal first is sent to the Region of interest (ROI) detection that is used to find where the binary squares are located and map the right square to the corresponding digit, resulting in separate vectors for each digit. These are individually normalized and digitized and then converted from a gray-code to the actual frame number together with the time index corresponding to the frame capture time. This set of data with frame number and time index for A , V , $\hat{A}$, and $\hat{V}$ is used to calculate the audio delay (A and $\hat{A}$), the video delay (V and $\hat{V}$) and the synchronization skewness ($\hat{A}$ and $\hat{V}$) with corresponding time index, using simple matching of frame number and difference calculation of corresponding time indices.

4.4 Reference signal

The reference signal is generated in MATLAB where the video stamps are added as an overlay of any video sequence, while the audio reference is generated as sequence of audio frames. The audio and video frames shown in Figs. 2 and 3 are then merged together in perfect alignment regarding frame number and time index, which is also verified. Any video sequence can be used for the video reference, in this case Ice. The choice of video sequence could impact the complexity of the video process and shall be based on the type of application that is tested, i.e. emulating a realistic scenario. Also, the video stamp squares should cover only a small amount of the video frame resolution to minimize impact on complexity of the video process.

4.5 Implementation of the method

In the implementation of the method both MATLAB and Python environment has been used and the setup is presented in Fig. 7, where the application to be tested is represented by the Sender and Receiver blocks. In the A/V Reference the audio and video signals are generated separately and merged in MATLAB, see Section 4.4, and since the same frame period and frame numbering is used for both signals this results in a fully synchronized reference stream. The A/V reference signal is played out to the Sender from a separate computer where the audio is electrically feed to the microphone input and the video is played out on a 60 fps display and captured by the Sender's camera. The data is sent over a network and rendered and played out by the Receiver. The A/V Capture Device is a 60 fps HD-DV camera with a microphone input. For video, the Capturing Device is set up so that both the received, $\hat{V}$, and the reference signal, V , are included side by side in the same captured frame. For

Fig. 7 The setup of the delay and synchronization test

capturing audio, the audio reference A is recorded on the left microphone channel while the received signal, $\hat{A}$, was recorded on the right channel. This recorded data was then fed into the Delay Sync Measure system. The Delay Sync Measure block in Fig. 7 was implemented in Python. Since the squares representing the positions of the marking of V and $\hat{V}$ are spatially fixed, in this implementation the position is marked manually before the processing starts. The reference signal is without skew, which was verified by using the reference both as input signal and degraded signal, so the decoded audio and video frame number together with referring time stamps can be used directly for calculating delay and skew. It should also be noted that skewness is only calculated for frames where both an audio and video number is detected, e.g. during a detected video freeze no skew result is generated. The limitation in the setup that should be taken into account is mainly video related: the refresh rate on the reference display, the shutter time of the camera at the sender side, the refresh rate of the receiver display, and finally the video shutter time of the A/V capture device. The display refresh rate used in this setup is 60 fps. These capturing limitations results in a maximum accuracy of 16.6 ms minus mis-alignment in the test setup chain, which make the impact of the audio setup limitations negligible considering audio sampling rate (F_s). Having $M = 8$ results in a maximum delay of 17 s which is then also the theoretical maximum skew detectable.

5 Test setup for proof of concept

In order to evaluate the proposed method the test setup in Fig. 7 was used, and a VoIP based audio and video two-way communication application with encrypted UDP data was chosen for the test. The factors impacting the delay and synchronization are many, e.g. packet losses, network jitter, compressions levels, CPU constraints, different internal delays in different devices and the OS. The most important conditions from the methods point of view is delay and audio/video skew together with packet loss and compression from which all impact factors can be derived.

To enable the test and verify it during realistic conditions, the application is running over a real network. This is carried out both with a transparent network and with a network with emulated packet loss. In addition to the artifacts generated by the packet loss as such, packet loss will also cause the application to decrease the bandwidth usage and thereby increase the compression. This results in the method being verified during realistic conditions, which means that the frame code information is going through all the signal processing steps, is sent over a network and finally played out and captured with real devices.

5.1 Test scenarios

To evaluate the proposed method five different test scenarios were constructed using both a real transparent network with no packet loss and a real network with emulated packet loss.

1. No added delay.
2. An artificial delay of 16 frames, which is approximately 1.07 s, is introduced after 20 s for both audio and video.
3. An artificial delay. After 20 s, a delay of 16 frames was added to video, and after 30 s audio is delayed with 31 frames, approximately 2.07 s.
4. A uniformly distributed 5 % packet loss is introduced, with no added delay. Resulting in increased compression.

5. A uniformly distributed 10 % packet loss is introduced, with no added delay. Resulting in increased compression.

The first scenario will address the delay of the application with low impact from the network. It should be noted that this metric not only gives the QoS timing coming from the network impact usually taken into account, but also the processing time for capturing and encoding at the sender time and for the decoding and rendering at receiver side. The second scenario will test that the audio and video measurements are aligned and that the method is not introducing any synchronization skewness. The third scenario, with different audio and video delay applied at different moments in time, will test the ability to measure the skewness. The fourth and fifth scenarios, with 5 and 10 % packet loss, respectively, will evaluate the robustness from the perspectives of packet loss as well as compression, since the increased packet loss rate will also result in an increased compressing level.

6 Results

All result comes from one application, but the method has been applied to and verified for several different applications. The method was applied several times for each of the different test scenarios. One representative session from each scenario is presented in Figs. 8, 9, 10, 11 and 12, where every calculated value is presented with a star in the graphs. It can be seen in the figures that the method is behaving in the way described and expected in Section 5.1. In Fig. 8 the scenario with no added delay, transparent network, is presented and thereby showing the application's audio and video delay together with its synchronization skewness with low impact from the network. These results will represent an estimate of the impact from accumulated processing time on sender and receiver side. In Table 2 the corresponding average values are presented.

It can be seen in Table 2 that there is an overall longer delay of the video than the audio, resulting in a positive synchronization skewness of $\approx 57\text{ms}$ which according to the recommendations is an acceptable skewness. In Fig. 9 it is shown that the audio and video measurements are aligned when the delay is applied, visualized with a stable synchronization skewness. In Fig. 10 when the delay is applied at different points and with different amounts it can be seen that the method can also detect this, resulting in skewness. Further, it can be seen that the delay in Figs. 9 and 10 is the accumulation of the delay from Fig. 8 and the artificial delay as expected. The maximum accuracy of 16.6 ms can also be seen in the video delay figures by the quantized delay values. In Figs. 11 and 12 packet losses are applied which is also observed by the decreased amount of correct detected frames. For video, packet loss results in lost frames which leads to freezes, this is indicated in the figures by a linearly increasing video delay. This is noticeable for the 10 % packet loss case. For audio, the packet loss leads to a lower number of detected frames. It can also be seen that the audio delay variation continues to be small even in the 10 % packet loss case which indicates that the variation is handled by the audio jitter buffer. From the figures it can also be seen that as long as an audio and video frame is successfully decoded with the same frame number a synchronization value can be calculated. This makes the method robust since only one frame, in this case representing 1/15s, needs to pass the whole system in order to be able to compile a new measurement, which is beneficial in demanding situations.

Fig. 8 Transmission over transparent network. *Top:* audio delay. *Center:* video delay. *Bottom:* Synchronization skewness

Fig. 9 Transmission over transparent network with artificial delay. The audio and video delay is added simultaneously. *Top:* audio delay. *Center:* video delay. *Bottom:* Synchronization skewness

Fig. 10 Transmission over transparent network with artificial delay where the delays are added separately, first audio and then video. *Top*: audio delay. *Center*: video delay. *Bottom*: Synchronization skewness

Fig. 11 Transmission over network with 5 % packet loss. *Top:* audio delay. *Center:* video delay. *Bottom:* Synchronization skewness

Fig. 12 Transmission over network with 10 % packet loss. *Top:* audio delay. *Center:* video delay. *Bottom:* Synchronization skewness

Table 2 Audio and video delay and synchronization for transmission over transparent network, see Fig. 8

Measured	Avg. [ms]	Std [ms]
Audio delay	119.7	0.65
Video delay	176.7	11.1
Synchronization	57.0	11.1

7 Conclusions

In the area of quality of experience (QoE) for audio and video streaming and two-way audio and video communication the skewness in audio and video synchronization together with the audio and video delay have a major importance. A novel method for measuring both synchronization and delay, without need of access to application internal data, has been presented. The solution consists of an audio and video reference stream, where audio and video frames are marked with frame numbers, which are decoded on the receiver side to enable calculation of synchronization and delay. The audio stamps consists of a sum of sinusoidal base functions while the video stamp is a binary pattern embedded into the frame. The method has been verified with a real two-way communication application in a transparent network where the results show that both delays introduced by the application and the network as well as synchronization skewness can be detected. Additional artificial delays were added to verify that the method was able to detect and behave correctly in these controlled test scenarios. To verify robustness against packet losses both 5 and 10 % packet loss levels were applied and it was shown that both audio and video delays were detected successfully. Further, the test system can be adapted to be used in several ways depending on whether the requirements come from a streaming service or a two-way communication application. When only the skew is of interest, no reference signal is needed for the calculation at the receiver side which makes it useful for streaming applications. The method presented opens up opportunities to build and design tests for many test scenarios, enabling evaluation of many kind of applications running on any platform, OS and device, e.g. for all applications developed for tablets and smartphones, under any conditions.

8 Future work

As stated in the conclusions the method can be used in many kinds of scenarios, but it can also be adapted for in-services usage. The method can be further developed by increasing the precision depending on the user scenario being requested, e.g. increasing the frame rates of capturing and reference rendering. The method will also be used for evaluation of new and upcoming WebRTC solutions regarding their robustness against packet loss as well as fluctuating network conditions.

Acknowledgments Thanks are directed to Ph.D. Sylvain Tourancheau for well considered inputs and helpful discussions.

References

1. Blakowski G, Steinmetz R (1996) A media synchronization survey: reference model, specification, and case studies. *IEEE Sel Areas Commun* 14(1):5–35

2. Boyaci O, Forte A, Baset S, Schulzrinne H (2009) vDelay: a tool to measure capture-to-display latency and frame rate. In: 11th IEEE international symposium on multimedia, 2009. ISM '09, pp 194–200
3. Claesson I, Rossholm (formerly Nilsson) A (2003) GSM TDMA frame rate internal active noise cancellation. *Int J Acoust Vib* 8:159–166
4. ETSI ETR 297: Human factors (HF); Human factors in videotelephony (1996)
5. ETSI TR 102 643: Human factors (HF); Quality of experience (QoE) requirements for real-time communication services (2010)
6. Hollier MP, Rimell AN, Hands DS, Voelcker RM (1999) Multi-modal perception. *BT Technol J* 17:35–46
7. Huang Z, Nahrstedt K, Shu L, Steinmetz R (2013) Evolution of temporal multimedia synchronization principles: a historical viewpoint. *ACM Trans Multimedia Comput Commun Appl (TOMCCAP)* 9(34):40–47
8. IETF RFC 3550: RTP: A transport protocol for real-time applications (2003)
9. ITU-R BT.1359-1: Relative timing of sound and vision for broadcasting (1998)
10. ITU-T: Recommendation J.100: - Tolerances for transmission time differences between vision and sound components of a television signal (1990)
11. ITU-T Series G: Recommendation G.1010: End-user multimedia QoS categories (2001)
12. ITU-T Series G: Recommendation G.114: One-way transmission time (2003)
13. ITU-T Series H: Audiovisual and multimedia systems, Supplement 1 (05/99) Application profile - sign language and lip-reading real-time conversation using low bit-rate video communication (1999)
14. Jansen J, Bulterman DCA (2013) User-centric video delay measurements. In: Proceeding of the 23rd ACM workshop on network and operating systems support for digital audio and video. ACM, pp 37–42
15. Kryczka A, Arefin A, Nahrstedt K (2013) AvCloak: a tool for black box latency measurements in video conferencing applications. In: IEEE international symposium on multimedia (ISM), pp 271–278
16. Liu Y, Sato Y (2008) Recovering audio-to-video synchronization by audiovisual correlation analysis. In: 19th international conference on pattern recognition, 2008. ICPR 2008, pp 1–4
17. Proakis JG, Manolakis DG (2007) Digital signal processing: [principles, algorithms and applications], 4th edn. Pearson Prentice Hall, Upper Saddle River
18. Raake A, Schoenenberg K, Skowronek J, Egger S (2013) Predicting speech quality based on interactivity and delay. In: Proceedings of the INTERSPEECH 2013. Lyon, pp 1549–1552
19. Radhakrishnan R, Terry K, Bauer C (2008) Audio and video signatures for synchronization. In: 2008 IEEE international conference on multimedia and expo, pp 1549–1552
20. Savage C (1997) A survey of combinatorial gray codes. *SIAM Rev* 39(4):605–629
21. Shen Z, Luo J, Zimmermann R, Vasilakos A (2011) Peer-to-peer media streaming: insights and new developments. *Proc IEEE* 99(12):2089–2109
22. Steinmetz R (1996) Human perception of jitter and media synchronization. *IEEE Sel Areas Commun* 14(1):61–72
23. Winkler S, Mohandas P (2008) The evolution of video quality measurement: from psnr to hybrid metrics. *IEEE Trans Broadcast* 54(3):660–668
24. Yamagishi K, Hayashi T (2005) Analysis of psychological factors for quality assessment of interactive multimodal service. In: *Electronic imaging*, vol 5666, pp 130–138
25. You J, Reiter U, Hannuksela MM, Gabbouj M, Perkis A (2010) Perceptual-based quality assessment for audiovisual services: a survey. *Signal Process Image Commun* 25(7):482–501
26. Zhou L, Chao HC, Vasilakos A (2011) Joint forensics-scheduling strategy for delay-sensitive multimedia applications over heterogeneous networks. *IEEE J Sel Areas Commun* 29(7):1358–1367
27. Zhou L, Xiong N, Shu L, Vasilakos AV, Yeo SS (2010) Context-aware middleware for multimedia services in heterogeneous networks. *IEEE Intell Syst* 25(2):40–47

Andreas Rossholm received his M.Sc. in Electrical Engineering with emphasis on Signal Processing and Telecommunications in 2000, and his Tech. Lic. in Applied Signal Processing in 2006 from Blekinge Institute of Technology. He is currently pursuing the Ph.D. degree at the department of Applied Signal Processing at the same university and also has a position at Microsoft's Skype Division, Stockholm, Sweden.

His research interests started out with audio and video signal processing including pre- and post-processing, mainly focusing on mobile devices with limited resources requiring efficient processing. He is now focused on the area of audio and video perceptual quality in communication systems and the factors influencing the quality, where his latest research work is performed.

Benny Löfström received his M.Sc.E.E. degree in 1983 and his Ph.D. degree in Signal Processing in 1992, both from Lund University.

In 1993 he was appointed Senior lecturer at Blekinge Institute of Technology, where he was also Head of department 1994-1999 and Dean of education 2002-2010. His current interest areas are signal processing, image processing and digital and wireless communications, and his research is focused on image and video processing for applications in mobile phones, and perceptual quality of image and video.